INTEGRATED MEMORY SYSTEM

PRIORITY CLAIM

[1] This application claims priority from European patent application No. 03425171.0 filed March 19, 2003, which is incorporated herein by reference.

TECHNICAL FIELD

[2] In its more general aspect, an embodiment of the present invention relates to the field of memory systems comprising at least a non-volatile memory and a controller.

[3] Such a system can be an independent information storage unit or it can be used as a control or processing unit for a general application.

[4] More particularly, an embodiment of the invention relates to an integrated memory system comprising at least a non-volatile memory and an automatic error corrector.

BACKGROUND OF THE INVENTION

[5] In this specific technical field, it is well known that in an information storage unit (**FIG. 2**) the controller functions to provide an interface for the user, according to a predetermined convention, by using an interface and some services made available by the memory.

[6] This is the typical condition of storage systems like disk-on-chip, memory card or smart card, as schematized in the here-attached **FIG. 2**.

[7] In a general control unit (**FIG. 3**) a predetermined software level attends to providing interface functions to overhanging software levels by using said interface and services made available by the memory.

[8] In an ideal system, without errors, the memory receives/sends the data sent/required by the controller without error possibility.

[9] In a real system, however, it is almost impossible to send and store data without errors.

[10] Data-storage flash memories use the most developed technologies to integrate the most information per area unit. For these applications NAND, Multilevel NAND and Multilevel NOR memories are commonly used. Error probability of these memories increases as the memory ages because of well-known physical phenomena.

[11] For example, in a floating gate memory cell, the information storage capacity is influenced by several mechanisms, such as: the ability to maintain charge, the "read disturb" mechanism, the "program disturb" mechanism, the SILC phenomenon, each having a different influence as the memory ages.

[12] In substance, the error tolerance which must be offered by the memory system is determined by the kind of application required by the user and it is usually lower than the error tolerance offered by flash memories during their life span.

[13] A convenient memory system must provide some devices and functions to show externally, or at overhanging levels, an error tolerance being lower or equal to the one required by the application. This is commonly achieved by using error correction techniques.

[14] These error correction techniques provide that the information is split in words of K bits and that these words are conveniently coded with N bits, where R redundancy bits are added to the K information bits. Only at this point are words written in the memory.

[15] During the information reading, the K original information bits are decoded from the N bits read by the memory.

[16] The presence of R redundancy bits allows the original information to be recovered if the error number is lower than W. Different codings for the K information bit protection from W errors by coding on N bits are known in the prior art.

[17] For convenience of illustration, a practical case of a linear and systematic coding (144, 128) for correcting two errors (N=144 bits, K=128 bits, W=2 bits) will be considered hereinafter.

[18] With reference to **FIG. 6**, the coding and decoding of such a code can be summarised as follows:

The vector "***c***" to be stored, composed of 144 bits, is obtained as a vector product of "***i***", a 128-bit data vector, with the binary generating matrix "G" composed of 128 rows by 144 columns. In the particular linear and systematic coding case G is composed of an identity matrix I and a parity matrix P. The vector "***c***" being obtained thus comprises the repetition of the data vector "***i***" and the parity vector is obtained as vector product iP. Formally expressed:

$c=i \cdot G=i \cdot [I,P]=[i, iP]$

[19] The matrix P is a binary matrix of size 128 X 16 being conveniently chosen (a method for obtaining this matrix is known in the art).

[20] For the decoding step a so-called syndrome "***s***" must be calculated:

$s=r \cdot H^T=p \oplus d \cdot P$

where

$G \cdot H^T=0$

[21] The syndrome "***s***", a 16-bit vector, is obtained as a vector product between the vector "***r***", a data vector of 144 bits read by the memory and composed of a 128-bit data part "***d***" and a 16-bit parity part "***p***", and the transpose of the matrix H. It can be shown that the syndrome is equivalent to the sum in GF(2) of the parity vector "***p***" with the vector product of the vector ***d*** with the parity matrix P.

[22] The syndrome s is void if the data vector "***r***" is correct; while, according to the way the matrix P is defined, "s" has a sole value for each possible configuration of one or two errors affecting "***r***".

[23] To achieve this coding and decoding scheme some solutions are known, which are schematically shown in **FIGS. 4a** and **b**.

[24] In the first solution of **FIG. 4a** a general memory is used. This memory stores data "***i***" and parity bits "***iP***" calculated by the controller. The coding and decoding task for correcting errors is entrusted to the controller.

[25] The disadvantages of this solution are:

- high latency time between the data request and the availability thereof.

[26] In fact standard flash memories use a limited-size interface bus to communicate with the controller. Although the memory internal reading is commonly

performed for pages having several bits (for example 144 bits), the communication towards the controller is bound to a word sequence on the data bus. Since for calculating the syndrome, and thus for detecting and correcting the error, all the page bits are necessary, data are available for the user only after transferring all K bits of the page (data + parity) in the controller and after calculating the parity.

- a memory overhead in the controller.

In fact the K bits of the page (data + parity) must be loaded in the controller memory with subsequent occupation of a controller memory area.

- an overhead in the controller processing resources.

[27] In fact the controller must, for each data page, read all K bits of the page (data + parity). For each page it must therefore calculate the parity and, if the latter indicates an error, activate the correction.

[28] For performing these operations the controller requires a calculation power overhead so as not to influence the memory system overall performances; moreover a part of the transmission band between the memory and the controller is used for transmitting parity bits being not strictly necessary for the user application.

[29] In this second prior art solution, shown in **FIG. 4b**, the coding and decoding task is entrusted to the memory. The syndrome decoding for correcting two errors is generally too onerous in terms of area to be advantageously implemented in the memory.

[30] The technical problem underlying an embodiment of the present invention is to provide an integrated memory system, having such structural and functional characteristics as to ensure an automatic error correction in data storage overcoming the drawbacks mentioned with reference to the prior art.

SUMMARY

[31] A solution underlying an embodiment of the present invention provides the possibility and opportunity to split the correction in different processes to be performed directly in the memory or in the controller.

[32] On the basis of this solution, the technical problem is solved by a memory system.

DESCRIPTION OF THE DRAWINGS

[33] The features and advantages of the system according to the invention will be apparent from the following description of at least one embodiment thereof given by way of non-limiting example with reference to the attached drawings.

- **FIG. 1** is a schematic block view of a prior-art memory system incorporating at least a non-volatile memory and a controller;
- **FIG. 2** is a schematic block view of the system of **FIG. 1** used as an information storage unit;
- **FIG. 3** is a schematic block view of the system of **FIG. 1** used as a control or processing unit;
- **FIG. 4a** schematically shows a prior-art memory system equipped with devices for the automatic error correction;
- **FIG. 4b** schematically shows a further prior-art memory system equipped with other devices for the automatic error correction;
- **FIG. 5** schematically shows a memory system according to an embodiment of the invention incorporating devices for the automatic error correction;
- **FIG. 6** is a schematic block view of a non-volatile memory portion incorporated in the system of figure 5 according to an embodiment of the invention;
- **FIGS. 7 to 10** are respective schematic block views of components of the memory of **FIG. 6** according to an embodiment of the invention.

DETAILED DESCRIPTION

[34] With reference to the drawings, and particularly to the example of **FIG. 5**, a memory system in accordance with an embodiment of the present invention to store data and information in binary logic is globally and schematically indicated with ***1***.

[35] Such a system can be an independent information storage unit (**FIG. 2**) or it can be used as a control or processing unit for a general application (**FIG. 3**).

[36] The system ***1*** comprises at least a non-volatile memory ***2***, for example of the Flash type, and a controller ***3*** associated to the memory **2** by means of an interface bus ***4***.

[37] Non-volatile memories are electronic devices integrated on a semiconductor and equipped with memory cell matrices, particularly multilevel cells.

[38] More particularly, memory device means any monolithic electronic system incorporating a memory cell matrix, organized into rows, referred to as word lines, and columns, referred to as bit lines, as well as circuit portions associated to the cell matrix and responsible for addressing, decoding, reading, writing and erasing the memory cell content.

[39] Such a device can be for example a memory chip integrated on a semiconductor and of the non-volatile EEPROM flash type split in sectors and electrically erasable. Each memory cell comprises a floating gate transistor with source, drain and control gate terminals.

[40] The memory system ***1*** is advantageously equipped with a particular device for the automatic correction of storage errors.

[41] In particular, the system ***1*** provides the possibility and opportunity to split the correction in different processes to be performed directly in the memory ***2*** or in the controller ***3***.

[42] As shown in **FIG. 6**, the structure of the memory system ***1*** is illustrated as a set of functional blocks located between an input terminal In and an output terminal Out.

[43] Memory ***2*** comprises, besides respective areas ***2a***, ***2b*** for data and parity storage:

- circuits COD for the coding required to correct two errors;
- a logic SYND for calculating the syndrome s;
- a circuit COR for correcting a single error; and
- a logic Ext COR for detecting more than one error.

[44] As illustrated, the memory ***2*** is coupled to a bus ***4*** in order to provide the following to the controller ***3***:

- a one-or-no-error-corrected data ;
- the uncorrected error;
- the calculated syndrome s;
- a signal IRQ activated to request the external correction.

[45] **FIGS. 7 to 10** show in schematic blocks the different blocks composing the memory ***2*** for a linear and systematic coding.

[46] **FIG. 7** shows the coding block COD located immediately downstream of the input terminal.

[47] The vector product ***iP***, performed by the block P, has a limited circuit complexity, proportional to the number of parity bits, and it is obtained through the synthesis of the corresponding logic function: c=[i, iP].

[48] **FIG. 8** shows the block SYND for calculating the syndrome s=p+dP.

[49] The calculation of the syndrome s uses almost integrally the parity calculation circuit used for the coding COD.

[50] **FIG. 9** shows the block COR for correcting one error and the block for requiring the external correction Ext COR with o=d+e.

[51] The block for decoding a single error includes a decoding circuit E, of limited complexity, proportional to the number of single errors to be corrected.

[52] This block is obtained through synthesis of the functional description thereof. In practise, the block E recognises each one of the 128 syndromes associated with a single error and it activates, through the vector "***e***", the correction of the corresponding bit.

[53] If there are no errors to be corrected (s=0) or the syndrome value does not belong to the set of single errors, the vector "***e***" is void and the original data **d** is output.

[54] **FIG. 10** shows a simple activation block of a signal ***q*** being simultaneously activated when the intervention of the external correction is required.

[55] The memory system **1** according to an embodiment of the invention, incorporating the previously mentioned correction blocks, uses the high parallelism available on the reading path. In fact, the parallel reading of a 144-bit data page and the writing, *i.e.*, the parallel programming of a 144-bit data page, are only available for memory-integrated coding, decoding and correction solutions.

[56] The operating method of the memory system according to an embodiment of the invention will now be described.

[57] A first error-correction process relates to the determination of the syndrome "s" and to the correction of single errors, performed in the memory ***2***.

[58] These functions are sufficiently simple from a circuit point of view so as to be advantageously performed in the memory ***2***.

[59] Since memory ***2*** simultaneously comprises all K bits of the read page (data + parity), the calculation of the syndrome s and the single-error correction can be immediately performed.

[60] This allows the corrected data to be transferred to the controller ***3***, immediately and with minimal latency, with no need to also transfer the parity. Therefore the bus band for the parity transfer is not occupied.

[61] If several errors occur, the memory ***2*** immediately indicates to the controller ***3***, through the dedicated IRQ line, the need to perform an external correction, in this case any alteration is applied on read data.

[62] The correction of all double errors, performed by the controller, will now be described.

[63] When, after a data request, the controller **3** receives from the memory ***2*** the indication that two or more errors occurred, the normal data transfer is interrupted.

[64] The syndrome ***s*** calculated in the memory **2** is read. If the syndrome is associated to a double error, the locations of the error-producing bits are detected, data are then read by the memory ***2*** and corrected where they must be corrected.

[65] This operation is not very frequent and more complex than the single error correction; in fact it requires more memory or calculation capacity, and can be advantageously performed by the controller without endangering system performances.

[66] Other particular cases will now be described:

1) The correction of more than two errors indicated by the syndrome ***s*** (Controller or user SW)

[67] If the syndrome calculated in the memory is not associated to any error, neither single nor double, it serves as a detector of three or more errors.

[68] The capacity of detecting all or a part of higher-order errors (3, 4, ...) exclusively depends on the chosen parity matrix. The detection of a higher-order error (*i.e.*, higher than ***2***) allows the controller ***3*** to use this information to activate the decoding of a correction code at a higher software level, on the same set of data or on a larger size set.

[69] The single errors being already corrected inside the memory and double errors being already corrected through the parity calculated in the memory, the events with three or more errors are extremely rare and, although the correction thereof is complex, they do not significantly affect system performances.

[70] The controller **3** overhead is limited to the correction of two errors, when explicitly required by the indication from the memory and it is limited to the reading of the syndrome ***s***, calculated in the memory, and to the correction of double errors.

[71] If the probability to have an error in a 128-bit page is for example 10^{-5}, the probability to have two errors is about 10^{-10}, the number of interventions by the controller ***3*** is therefore reduced by five quantities. The probability to have more than two errors is about 10^{-15} and thus extremely improbable.

[72] When required by the memory, through the IRQ signal, the controller provides to:

- read the syndrome ***s***
- calculate the two bits to be redounded through one of the following two methods:

searching, on an ordered table, the syndromes of two errors. This method requires an ordered table of the N(N-1) syndromes associated to two error-producing bits and a number of operations equal to O(Nlog N).

2) Comparing all the syndromes obtainable as sum of two rows of the parity P matrix.

[73] If the search detects the couple of error-producing bits, then the data reading, the correction thereof and the transmission to the user are performed.

[74] If the search does not detect the couple of error-producing bits, this means that the syndrome indicates the presence of three or more errors. The controller or driver SW asks the higher SW levels to correct these errors.

[75] The memory system ***1*** may be part of an electronic system, such as a computer system.

[76] It will be evident to the expert in the art how to apply the error-correction devices according to an embodiment of the present invention also to other circuit solutions allowing a split of the single error correction functions. For example the principles of an embodiment of the present invention can be applied also when the error-correction system is taken from the memory and integrated in a coprocessor or in an accelerator in the control unit.